

DRIVE PX 2 HANDS-ON

NVIDIA Automotive

Rev. 20170725

Drive PX 2 SDK 4.1.8.0

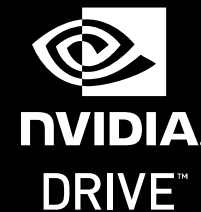
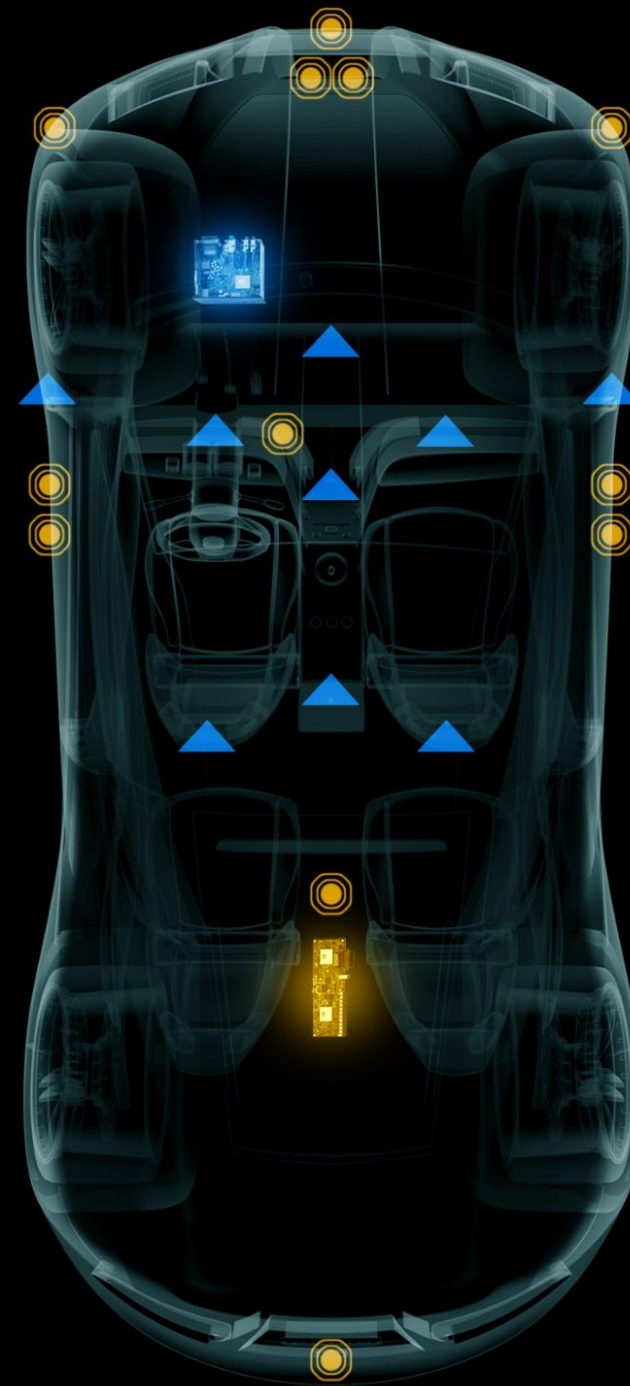


TABLE OF CONTENT

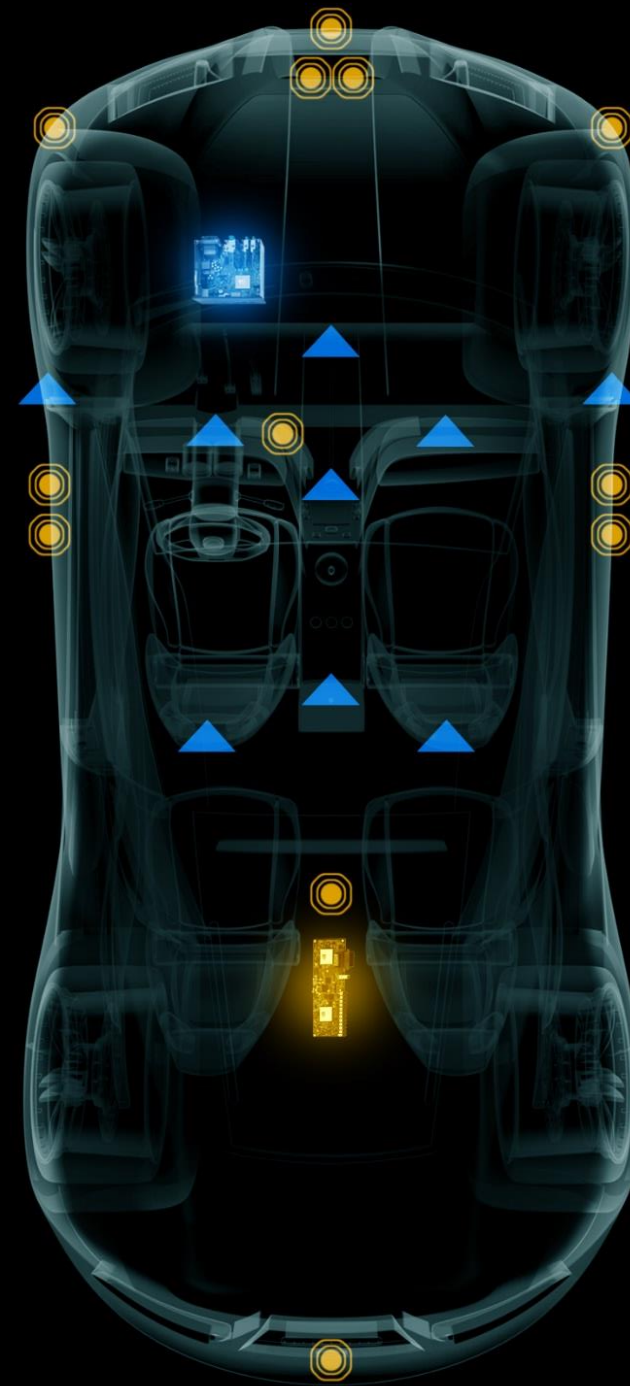
[DRIVE PX 2 Overview](#)

[DRIVE PX 2 Hardware Setup](#)

[DRIVE PX 2 SDK Installation](#)

[DRIVE PX 2 Hands-on](#)

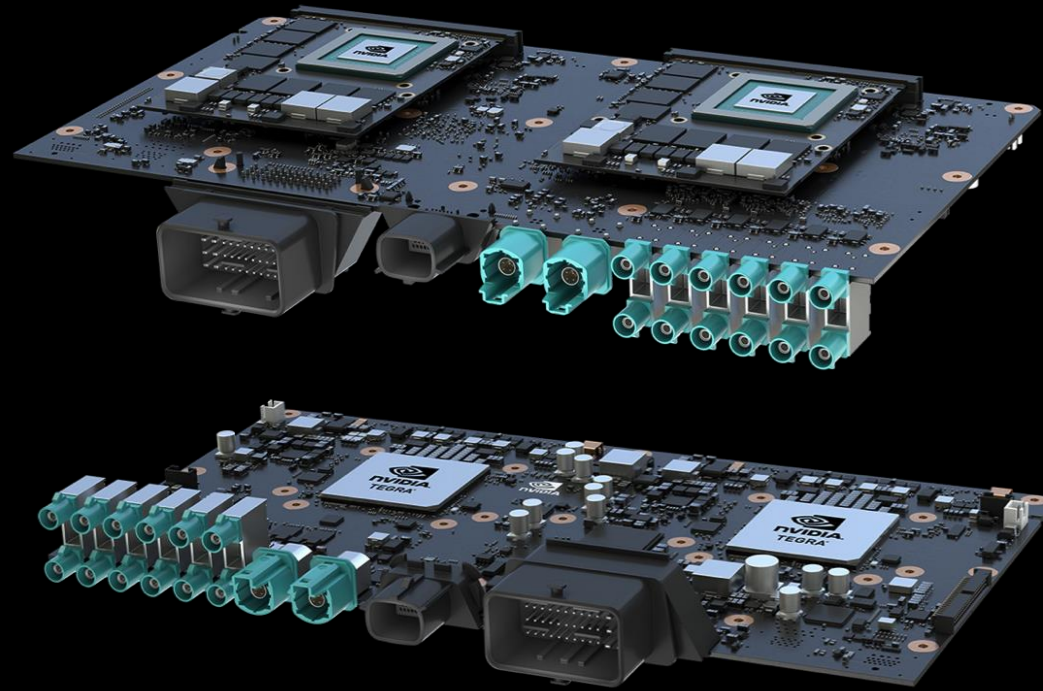
DRIVE PX 2 OVERVIEW



DRIVE PX 2

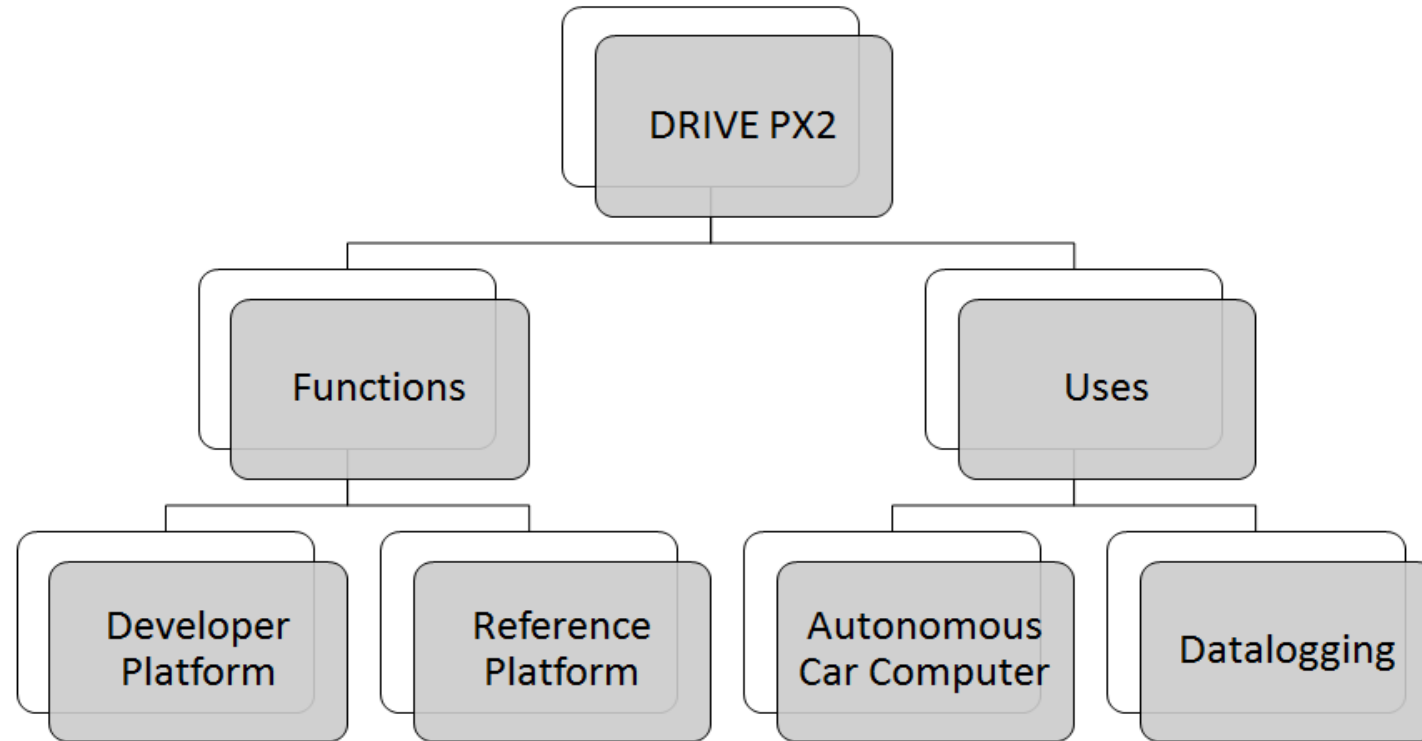
AutoChauffeur

- Scalable from 1 to 4 processors to multiple DRIVE PX 2s
 - 2x Tegra Parker SoC
 - 2x Pascal dGPU
 - 8 TFLOPS
 - 24 DNN TOPs
- Up to 12 cameras, plus LIDAR, radar, ultrasonic sensors
- DriveWorks SW/SDK
- AI perception
- Mapping



NVIDIA DRIVE PX 2

BUILT FOR



NVIDIA DRIVE PX 2 - SOFTWARE STACK

A full stack of rich software components

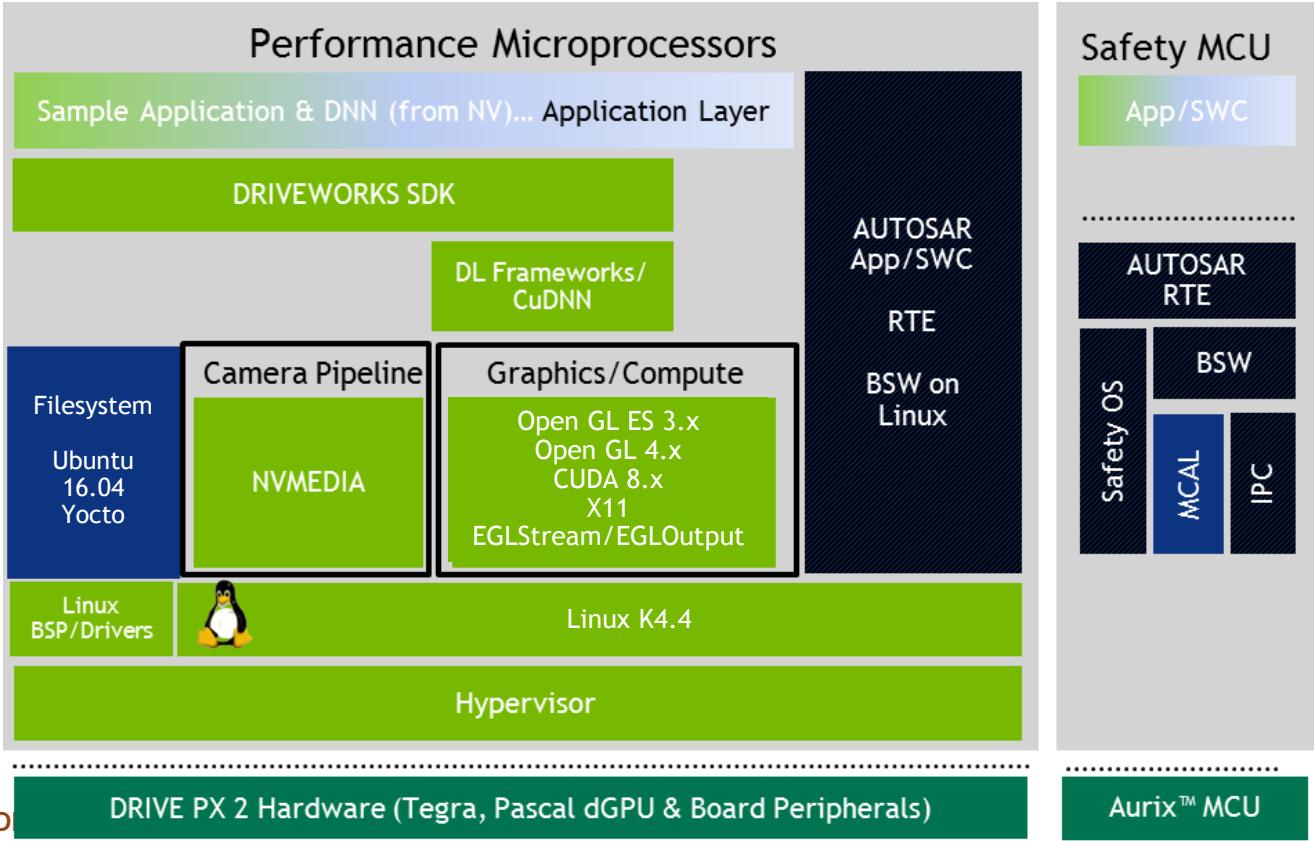
NVIDIA Vibrante Linux & Comprehensive BSP

Rich Middleware

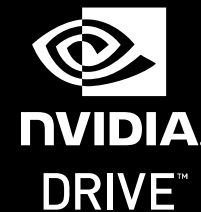
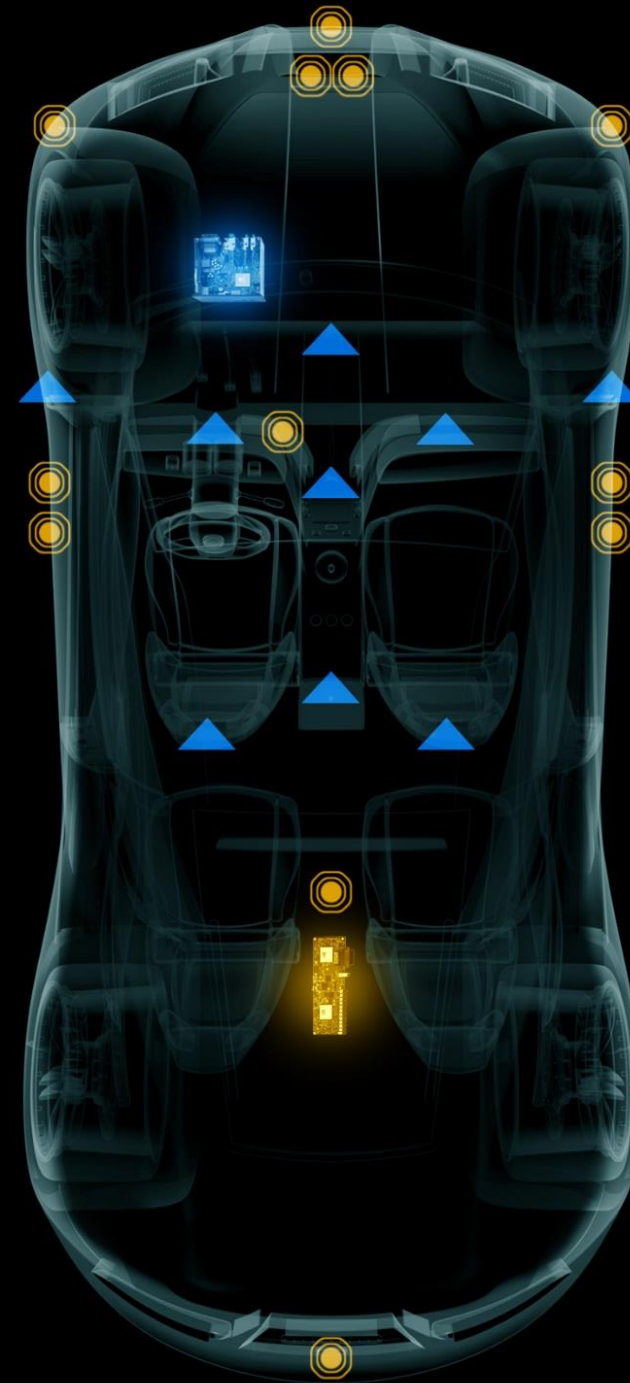
SDK, Samples and more

■ T1/OEM SW ■ OS/3rd SW/HW ■ NVIDIA Licensed SW ■ Drive PX Hardware
■ Elektrotbit

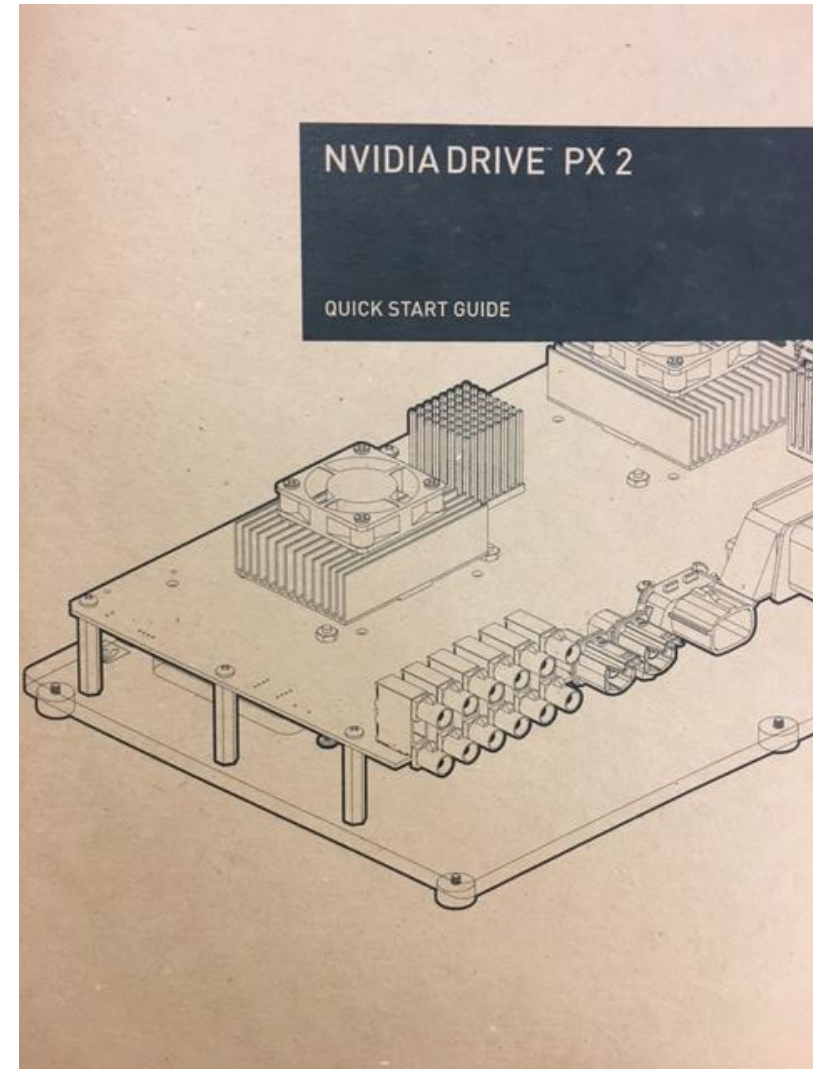
NVIDIA CONFIDENTIAL – D



DRIVE PX 2 HARDWARE SETUP



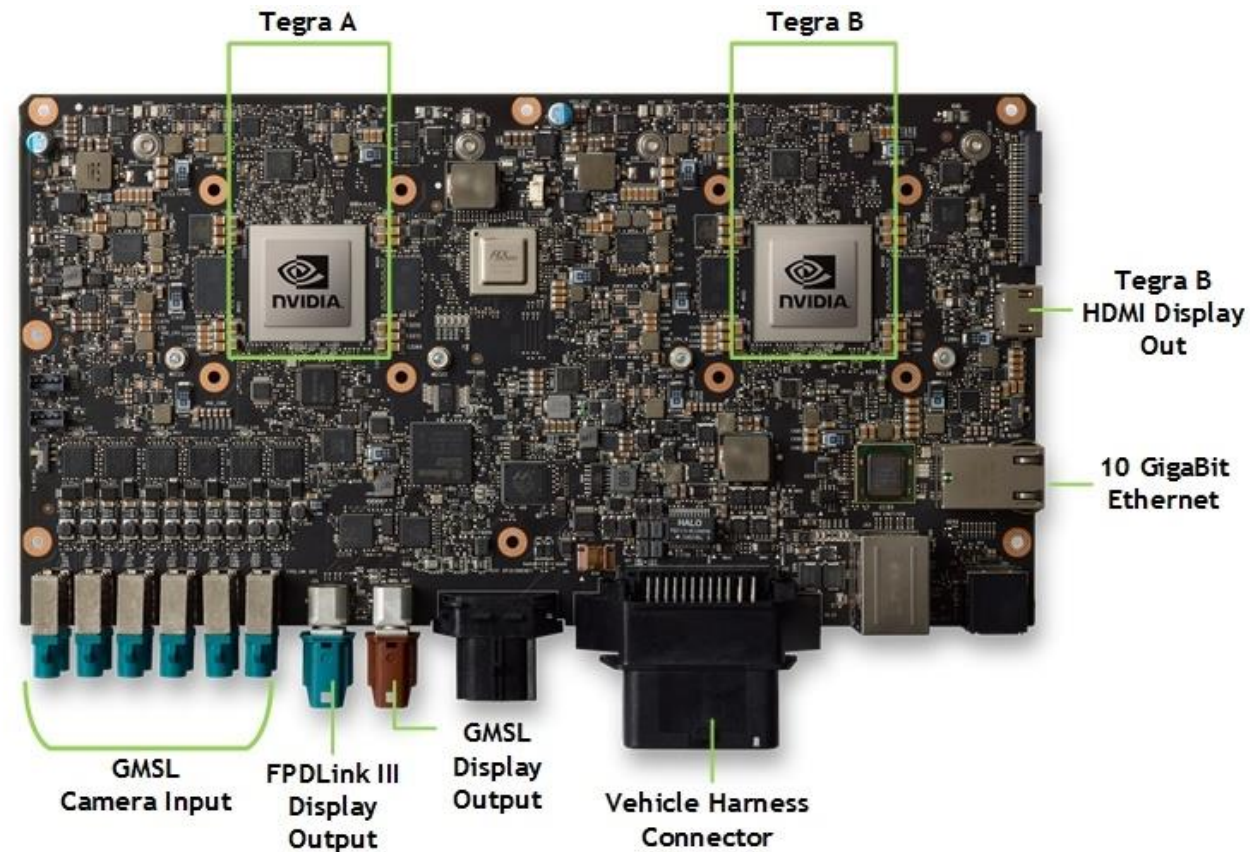
Check out the
Quick start guide!



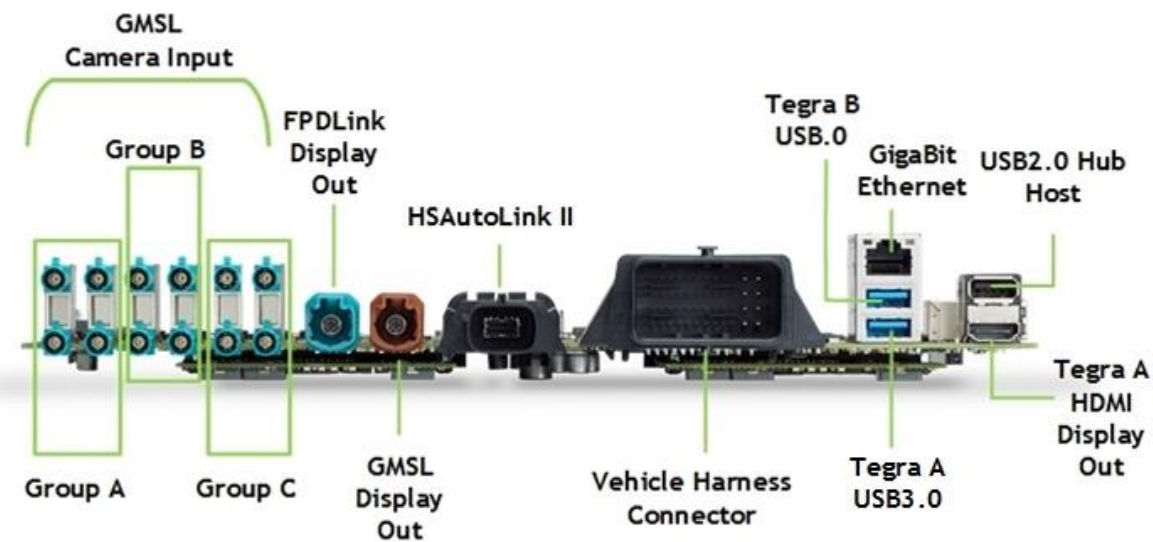
Included in your DRIVE PX 2 box

DRIVE PX 2 Connectors

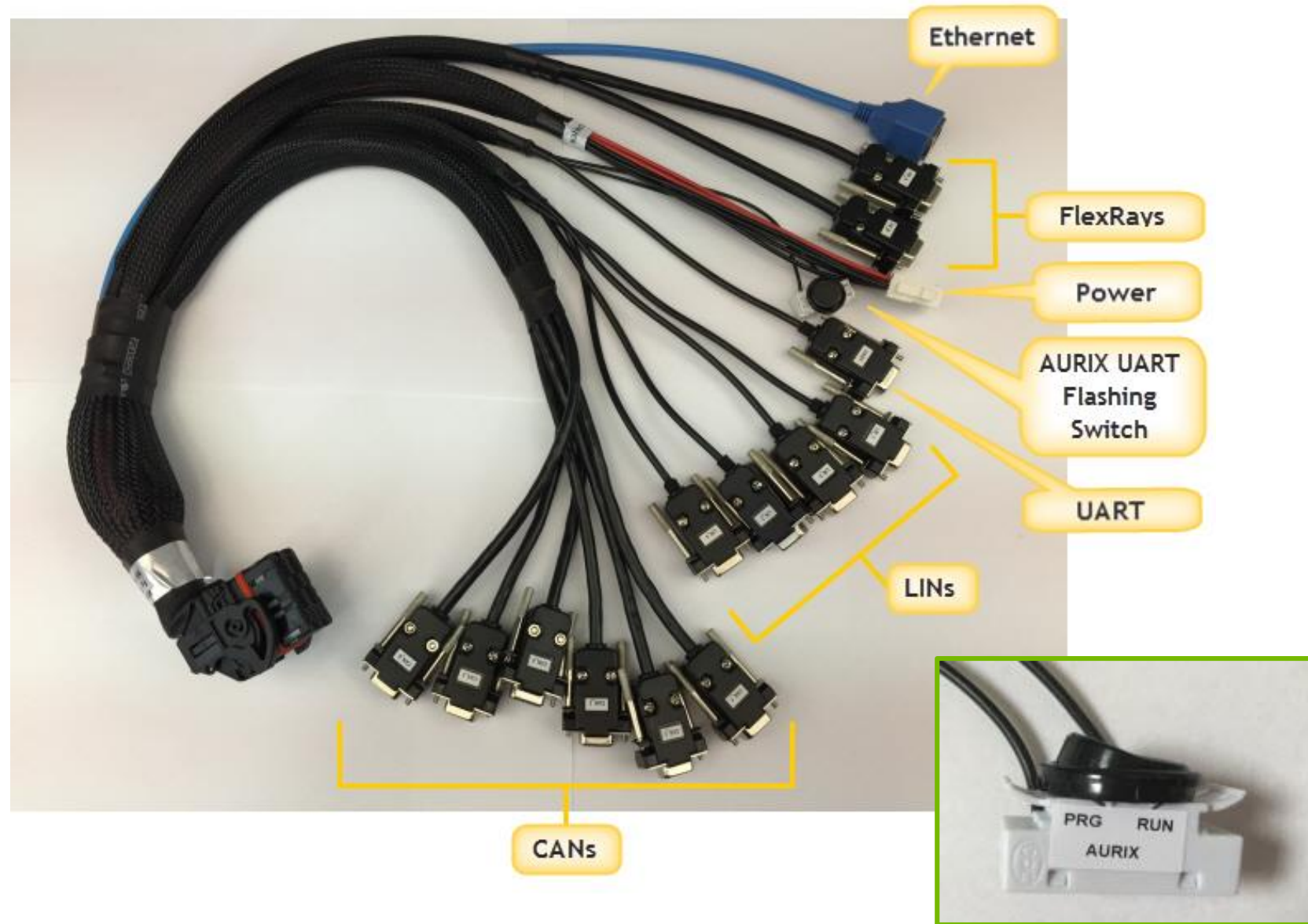
Top
View



Front
View



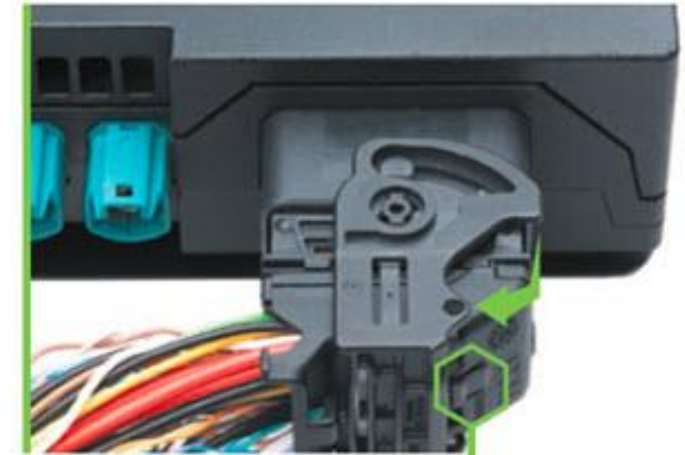
DRIVE PX 2 Harness Connectors



Note: Make sure that the AURIX Flashing switch is in the **RUN** position

Step 1

Vehicle Harness Cable



Must press to unlock the latch.

ATTENTION:

Do not plug in the power supply to the wall socket unless instructed.

- a) Connect vehicle harness
 - Plug the vehicle harness cable to DRIVE PX 2
 - Lock the DRIVE PX 2 Vehicle Harness
- b) Connect the Power Supply to the harness cable

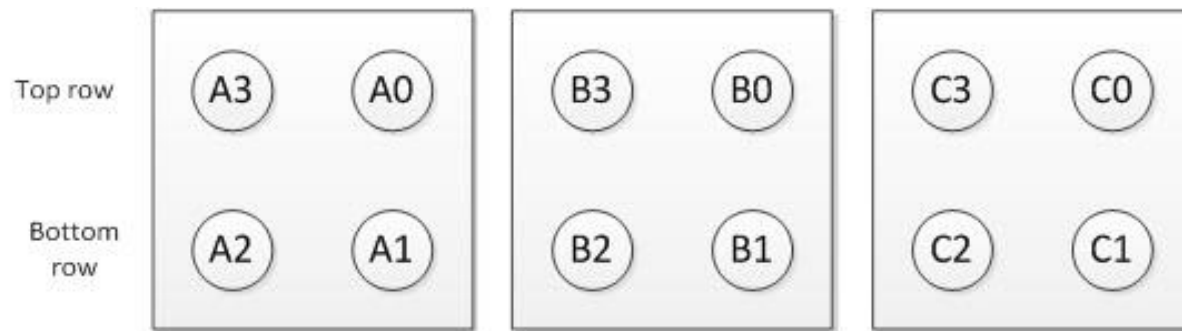


Step 2

Connecting the Cameras

If you don't have GMSL cameras, proceed to Step 3

Connect your GMSL Camera (e.g. AR0231) to DRIVE PX 2 at Port **A0**



ATTENTION:

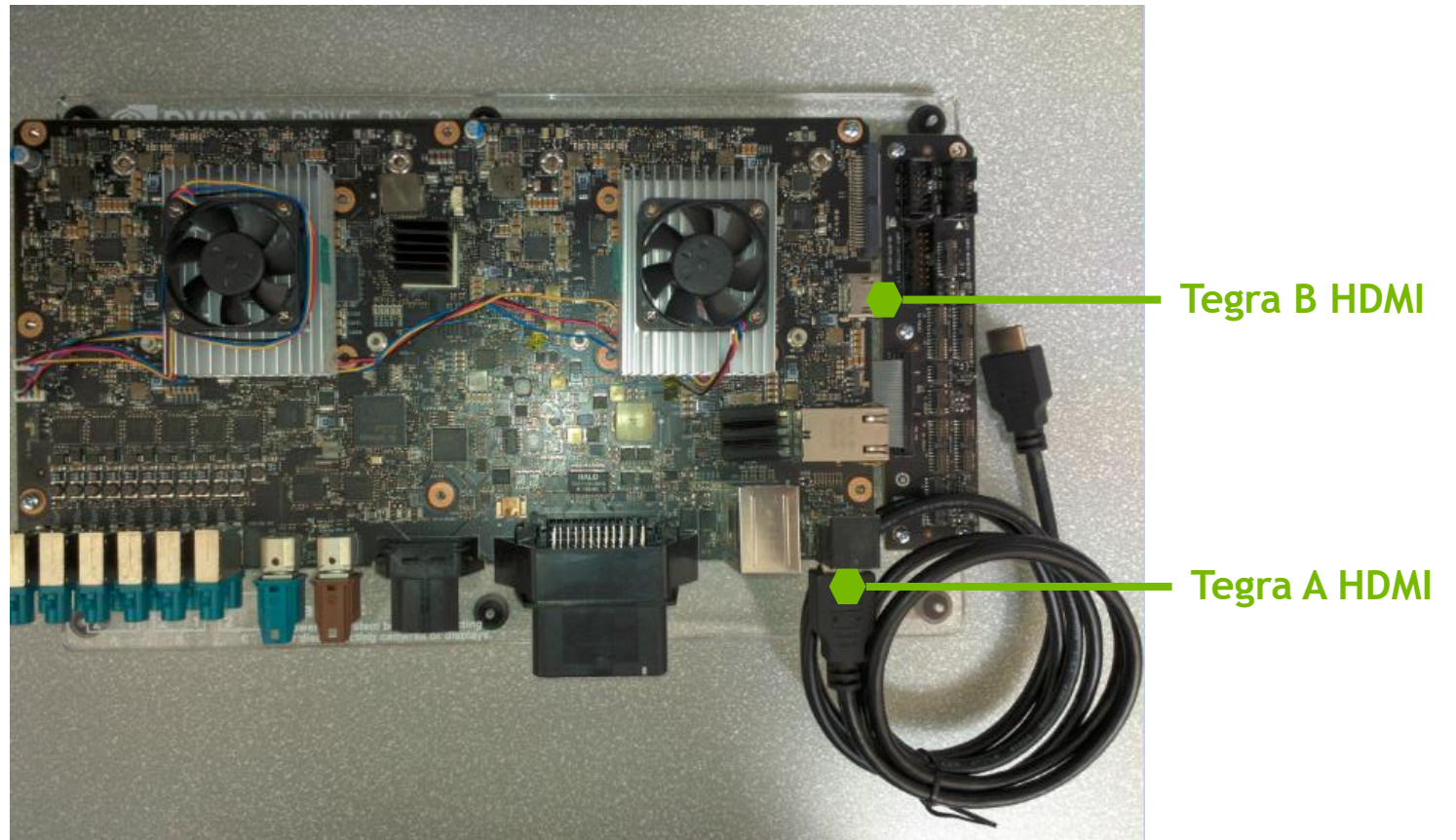
Always turn off main power before connecting or disconnecting cameras from NVIDIA DRIVE PX 2.

Note: If one camera is used, use the **A0** input. If more cameras are used, follow this order: **A0, A1, A2, A3**, then proceed with the next camera group in the same order: **B0, B1, B2, B3**, etc.

Step 3

Connecting the HDMI Display

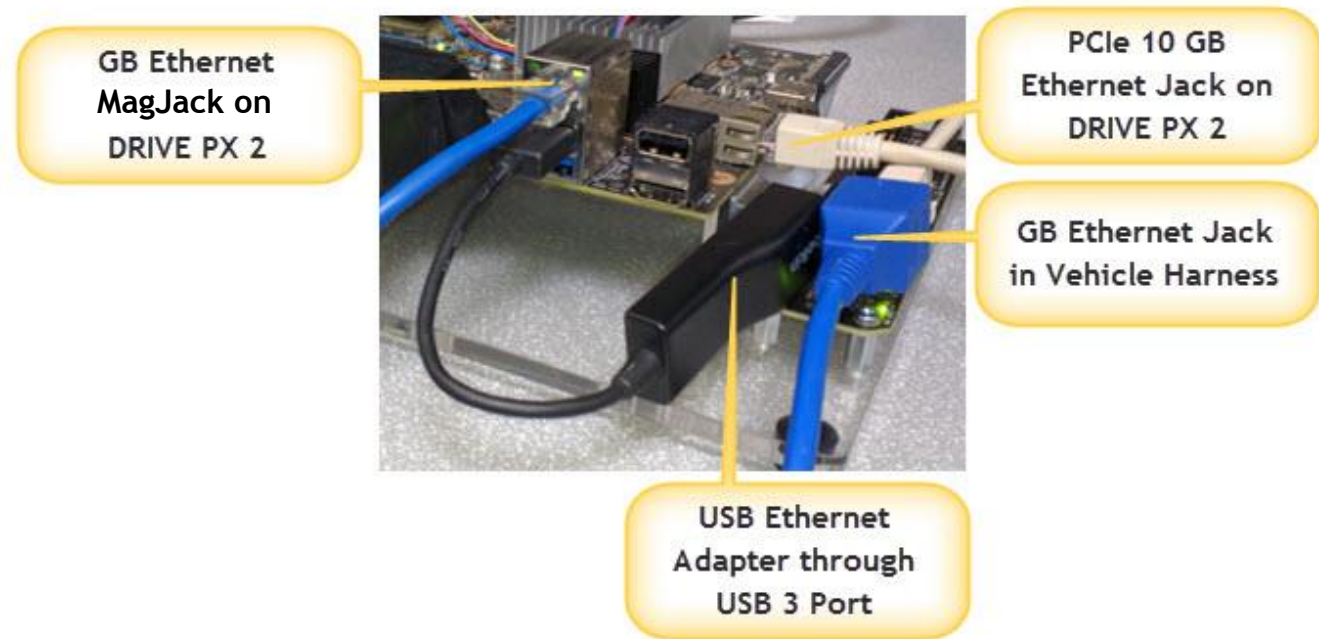
- a) Connect the HDMI Cable to Tegra-A HDMI port;
- b) Connect the HDMI Cable to Tegra-B HDMI port (optional);
- c) Connect the other end of the HDMI cable to display/monitor.



Step 4

Connecting to the Network

DRIVE PX 2 contains the following network interfaces to provide network connection for each Tegra Chip.

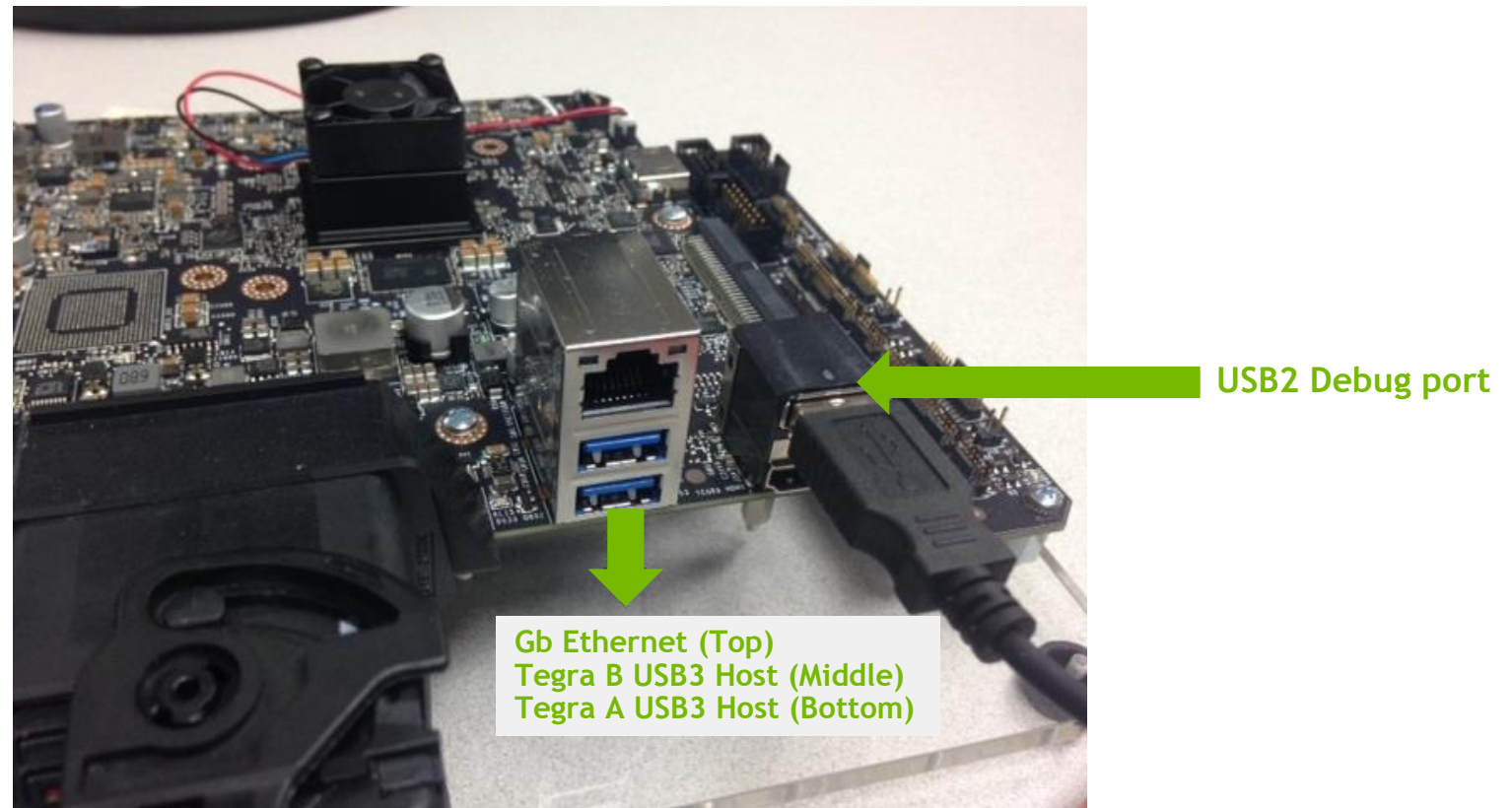


Connect one side of the Ethernet cable to the MagJack on DRIVE PX 2 or to the Ethernet Jack from the vehicle harness. Connect the other side to a wall Ethernet socket or a router that has broadband internet connection.

Step 5

Connecting the USB Cables

- a) Connect the USB A-A cable from the USB2 Debug port (on DRIVE PX 2) to your Ubuntu Host machine;
- b) Connect a keyboard & mouse to the provided USB Hub, and connect the Hub to the Tegra USB3 port.



Step 6

Plug the power supply to the AC Outlet

After plugging the power supply to an AC outlet, DRIVE PX 2 will boot. When logging in for the first time, a wizard will guide you through a few configuration steps (auto-login configuration, agreements acceptance etc.)



Note: If needed, the password for the account user `nvidia` is: `nvidia`

Try it!

The DRIVE PX 2 root filesystem is based on Ubuntu 16.04:

- System administration (users, networking, package installation etc.) is similar to usual Ubuntu system;
- **apt-get** provides access to all of the Ubuntu arm64 packages!

```
nvidia@tegra-ubuntu: ~  
nvidia@tegra-ubuntu:~$ sudo apt-get install vim  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  vim-common vim-runtime  
Suggested packages:  
  ctags vim-doc vim-scripts vim-gnome-py2 | vim-gtk-py2 | vim-gtk3-py2  
  | vim-athena-py2 | vim-nox-py2  
The following packages will be upgraded:  
  vim vim-common vim-runtime  
3 upgraded, 0 newly installed, 0 to remove and 338 not upgraded.  
Need to get 0 B/6,122 kB of archives.  
After this operation, 58.4 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

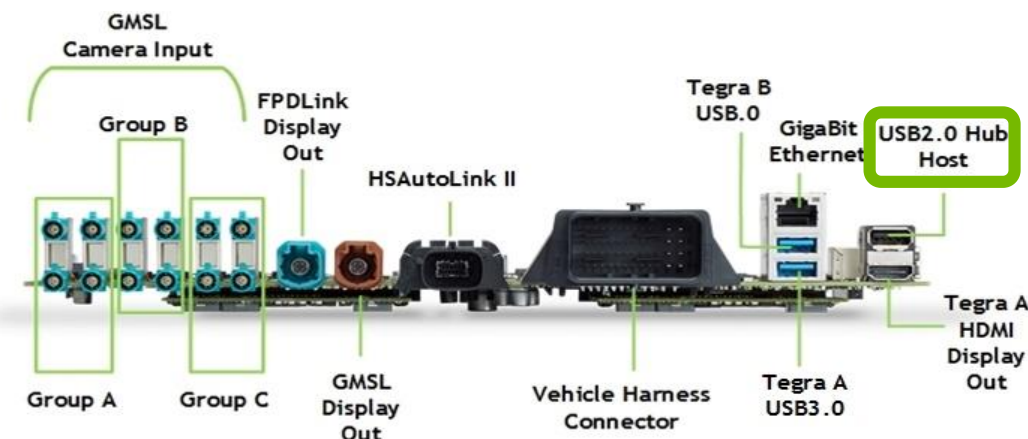
Accessing DRIVE PX 2 via the Debug UART

TEGRA/AURIX DEBUG UART

DRIVE PX 2 UARTs

The DRIVE PX 2 provides UART console for:

- Each Tegra & the AURIX MCU via a single USB A-A cable connected to the **USB2.0 Host Port** on the DRIVE PX 2 board. This is the preferred way to access the Tegra's or the AURIX console from Linux or Windows.
- AURIX MCU-only via the **UART connector on the DRIVE PX 2 harness**. This requires an USB-serial adapter. **Note:** when connecting via the UART harness connector, make sure that the USB A-A cable is **not** connected to the USB2 Debug port.



TEGRA/AURIX DEBUG UART

From Linux Host (1/2)

Requirements for the Linux Host system:

- Ubuntu 14.04 LTS (Trusty), AMD64
- Minicom software:

```
$ sudo apt-get install minicom
```

TEGRA/AURIX DEBUG UART

From Linux Host (2/2)

Minicom configuration on the host:

```
$ sudo minicom -s
```

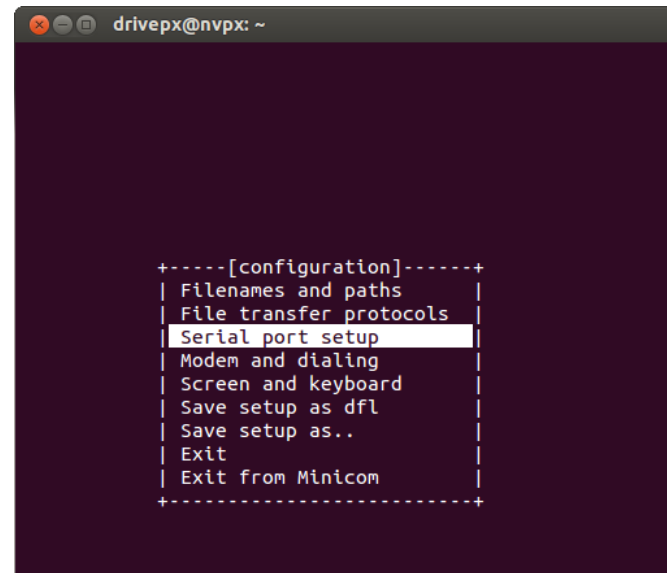
Select “Serial Port Setup”, configure with 115200 8N1, no HW & no SW flow control. Save the setup as the default.

Minicom can then be launched on a specific device with “-D” option. Use `ttUSB1` for AURIX, `ttUSB2` for Tegra A, `ttUSB6` for Tegra B:

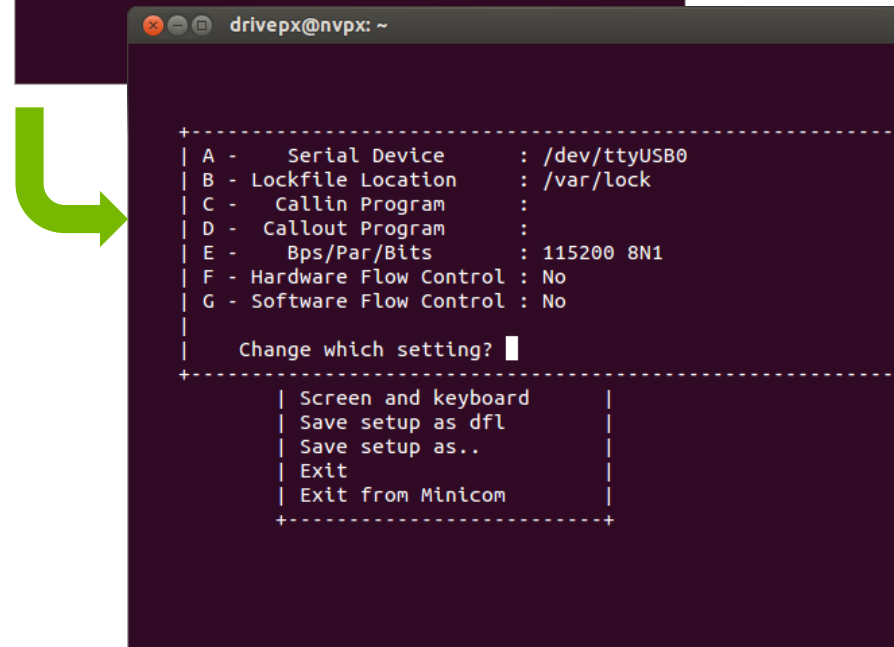
```
$ sudo minicom -D /dev/ttUSB2
```

Available serial devices can also be queried on the host with:

```
$ dmesg | grep ttUSB
```



```
drivepx@nvpx: ~  
+-----[configuration]-----+  
| Filenames and paths          |  
| File transfer protocols      |  
| Serial port setup           |  
| Modem and dialing           |  
| Screen and keyboard         |  
| Save setup as dfl           |  
| Save setup as..             |  
| Exit                         |  
| Exit from Minicom           |  
+-----+
```



```
drivepx@nvpx: ~  
+-----+  
| A - Serial Device           : /dev/ttUSB0  
| B - Lockfile Location       : /var/lock  
| C - Callin Program          :  
| D - Callout Program         :  
| E - Bps/Par/Bits            : 115200 8N1  
| F - Hardware Flow Control   : No  
| G - Software Flow Control   : No  
+-----+  
Change which setting?   
+-----+  
| Screen and keyboard         |  
| Save setup as dfl           |  
| Save setup as..             |  
| Exit                         |  
| Exit from Minicom           |  
+-----+
```

TEGRA/AURIX DEBUG UART

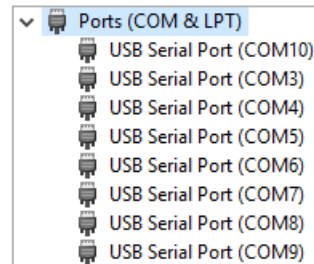
From Windows Host

On Windows, PuTTY can be used to connect to the DRIVE PX 2 UART:

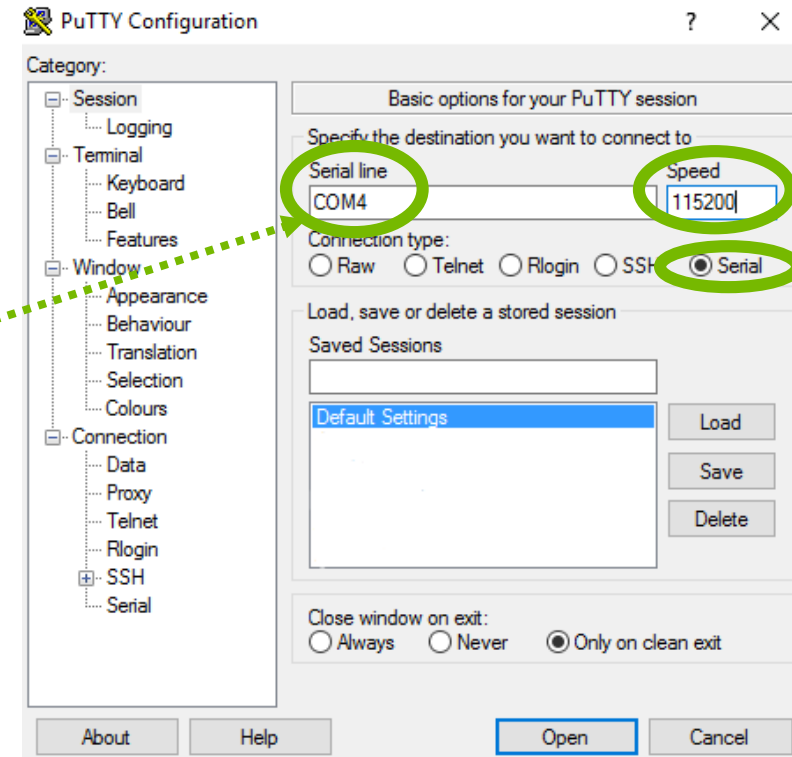
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Serial configuration:

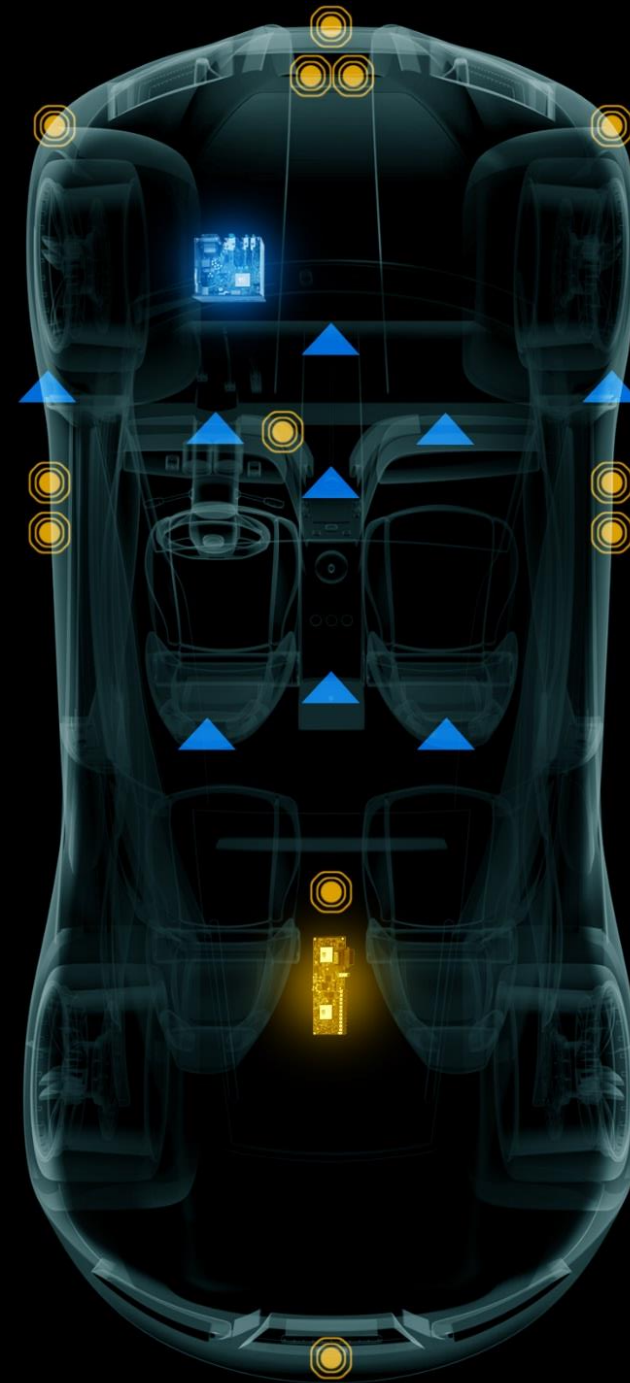
- Configure the speed to **115200**
- For the COM port number, check your Windows Device Manager:



- Try the various COM ports to find out which one is the AURIX or the Tegras.



DRIVE PX 2 SDK INSTALLATION



Pre-requisites

Requirements for the Host system:

- Ubuntu 14.04 LTS (Trusty), AMD64
 - Note: Do not upgrade the NVIDIA Drivers!
- NVIDIA GPU for CUDA host support
- 32GB of disk space
 - DriveInstall will automatically download the SDK delivery during the installation process
 - After unpacking and installation, ~8GB of disk space can be freed by removing the “7z” files in the DriveInstall folder.

NVIDIA DRIVE SDK

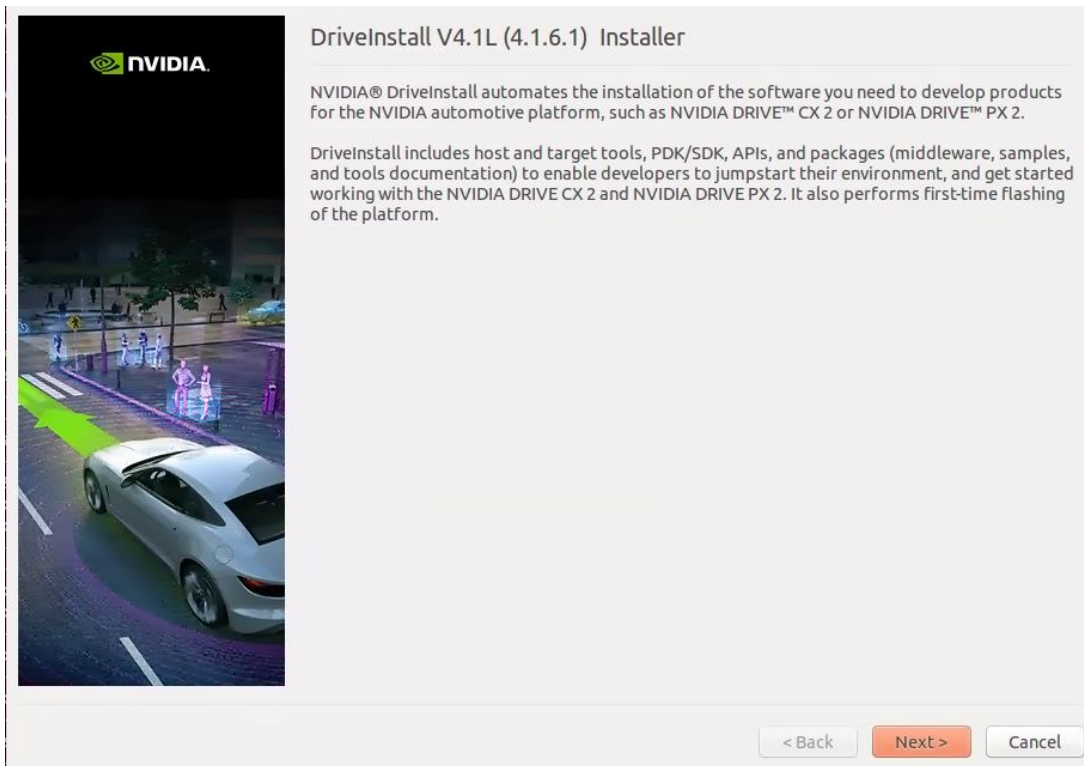
Download NVIDIA DRIVE SDK available from
<https://developer.nvidia.com/nvidia-drive-downloads>

The delivery consists of DriveInstall Linux SDK, which is an on-demand all-in-one package that bundles developer software for the NVIDIA® DRIVE Platform for Ubuntu 14.04.

It can be used to flash your DRIVE PX 2 with the latest OS image, to install developer tools for both host PC and target device, and to install the libraries and APIs, samples, and documentation needed to jumpstart your development environment.

SDK INSTALLATION

DriveInstall (1/4)



DriveInstall automates the Drive SDK installation as well as the Linux Host configuration necessary for the DRIVE PX 2. It can also flash the Tegras on the DRIVE PX 2.

SDK INSTALLATION

DriveInstall (2/4)

Pre-requisites:

- Your DRIVE PX 2 is connected to the Internet via the Gb Ethernet MagJack;
- Your Host PC runs Linux Ubuntu x64 14.04 or 16.04 LTS and is also connected to the Internet;
- The DRIVE PX 2 is connected to your host PC via the USB A-A debug cable. **All other USB-serial adaptors must be unplugged from the DRIVE PX 2;**
- Make sure to **EXIT ALL minicom terminal windows** connected to the DRIVE PX 2 device;
- On the host machine, download the DriveInstall application in a newly created folder, whose path **does not contain a period (“.”), nor any spaces.**
- Add exec permission and execute DriveInstall: from a terminal window, type

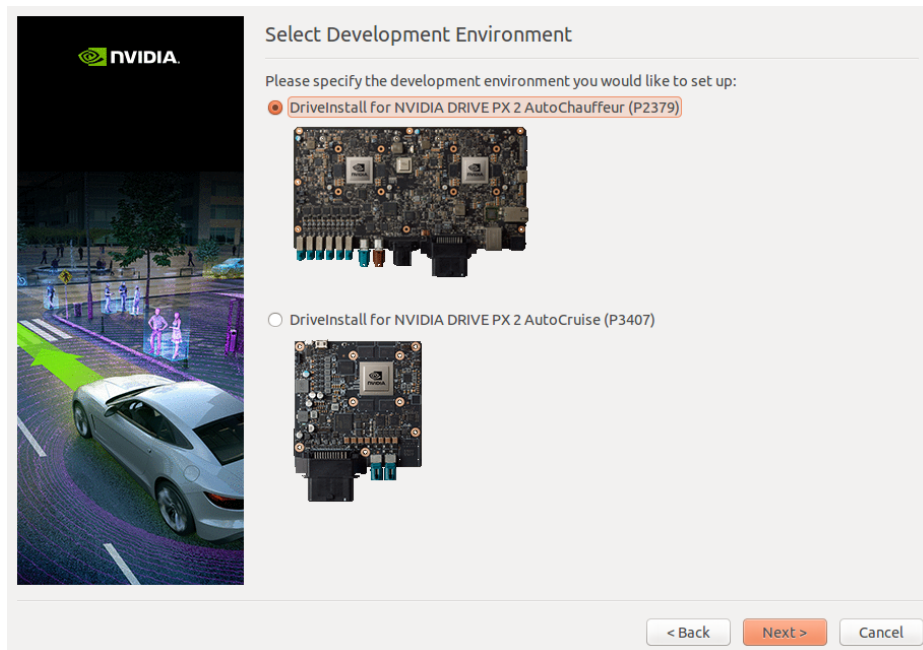
```
$ cd <path_where_DriveInstall_file_is_located>  
$ chmod +x DriveInstall-*.run
```

SDK INSTALLATION

DriveInstall (3/4)

Launch DriveInstall by executing its `run` file:

```
$ ./DriveInstall-*.run
```

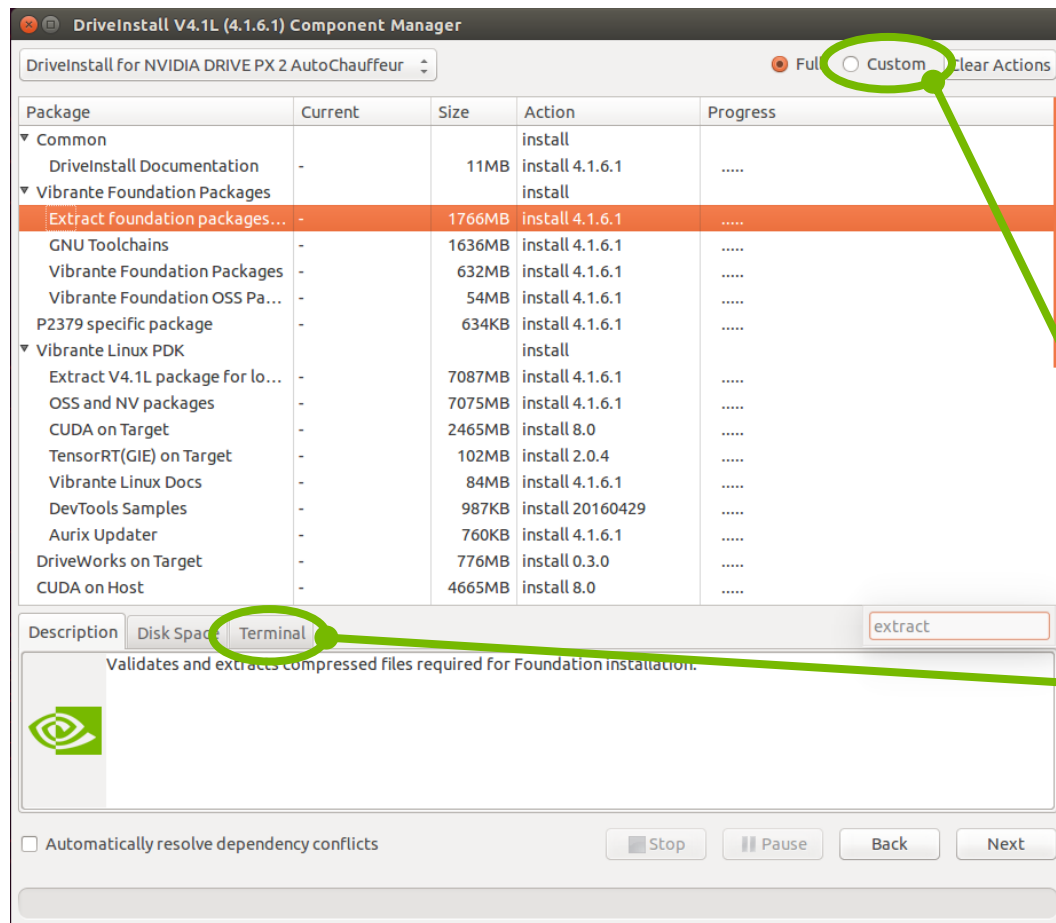


When asked, specify the installation directories and click `Next`.

After that, select **DRIVE PX 2** as the development environment.

SDK INSTALLATION

DriveInstall (4/4)



The DriveInstall component manager allows you to select what files you want to install. Choose the **Full** installation.

In case you only want to install the SDK on your Host PC and do not want to flash the DRIVE PX 2 Tegras, you can set the installation to **“Custom”** (on the top right) and the **“Flash Device”** action to **“no action”** (by left-clicking on the action field).

During the installation process, watch for the **Terminal** tab that will ask you for confirmations.

At the end of the installation, the SDK is installed in the **“VibranteSDK/”** directory.

SDK INSTALLATION

First boot

After the process is completed, you will then have an Ubuntu Desktop.



Note: If needed, the password for the account user `nvidia` is: `nvidia`.

DRIVE PX 2 HANDS-ON

AURIX Console

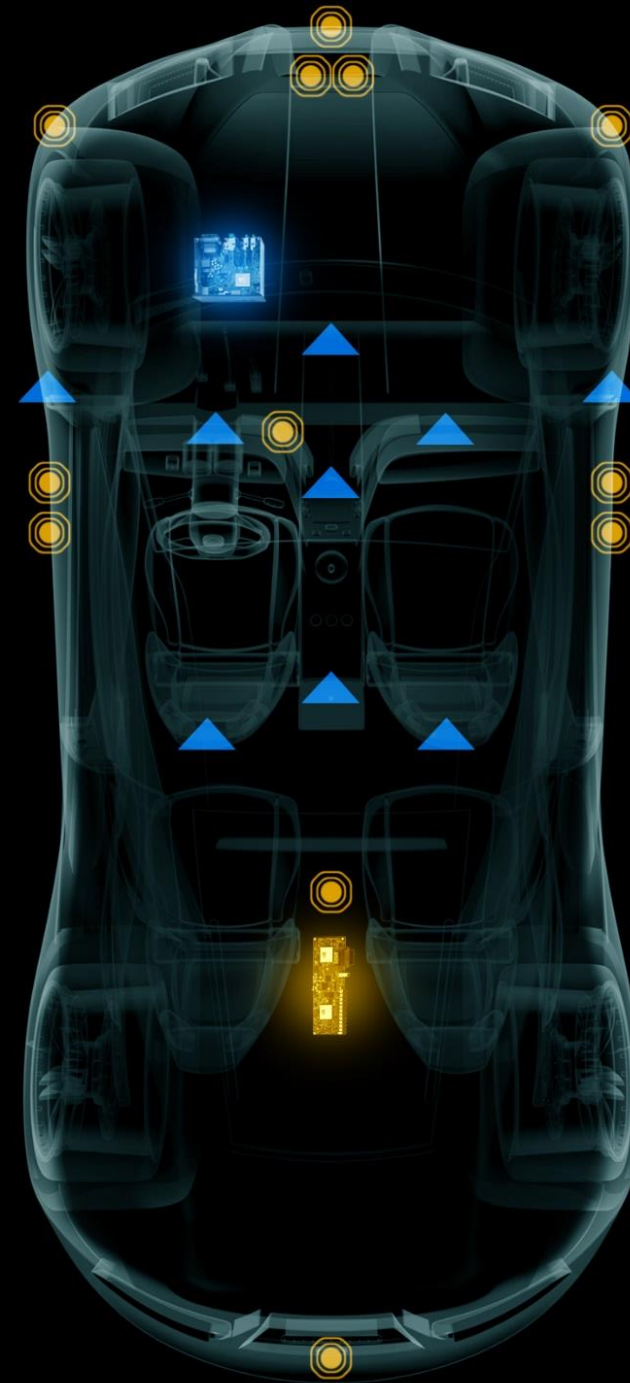
CUDA Sample Applications

DriveWorks Sample Applications

VisionWorks Sample Applications

Inter-Tegra communication

DRIVE PX 2 Tips



AURIX Console

AURIX CONSOLE

Setup

The AURIX console is available over UART via the USB debug port on DRIVE PX 2:

1. Connect the USB A-A cable from your DRIVE PX 2 USB debug port to Linux host;
2. Start a Host terminal program (e.g. minicom) and connect to the AURIX:
 - ✓ Device should be `/dev/ttyUSB1` assuming there is no other USB/Serial adapters connected to your Linux host
3. Power-on your DRIVE PX 2, and your terminal will provide a shell from the AURIX:



```
shell>
```


AURIX CONSOLE

Commonly Used Commands

Command	Description
help	Shows all available commands supported
status	Provides IP & MAC address of the AURIX & Tegra's
power [a b ab] [on off]	Power on/off the Tegra A/B/Both system, excluding AURIX
aurixreset	Power cycle the DRIVE PX 2 board, including AURIX
tegrareset [a b] [h]	Toggle reset Tegra A/B. Default is 'a'. If h is supplied then hold Tegra A/B in reset.
tegrarecovery [a b] [on off]	Set or clear Tegra A/B force recovery mode
campower [a b c] [on off z]	Turn on or off for the camera module on group A, B, or C; z configures the power control pin to high impedance
hdmiastource [tegra mxm]	Select Tegra A or MXM (dGPU) to source HDMI A
hdmibsource [tegra mxm]	Select Tegra B or MXM (dGPU) to source HDMI B
fpdlsorce [a b]	Select Tegra A/B as FPDLink display out source
version	Shows the SW version

CUDA Sample Applications

CUDA ON TEGRA

Target Setup & Sample Applications

Add the path to the CUDA binaries & libraries in your nvidia user `.bashrc` file:

```
export PATH=/usr/local/cuda/bin/:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/targets/aarch64-linux/lib:$LD_LIBRARY_PATH
```

You can install the CUDA samples in your home directory with:

```
$ /usr/local/cuda/bin/cuda-install-samples-8.0.sh ~/
```

The sample directory is called `NVIDIA_CUDA-8.0_Samples/`. For example, you can compile & execute the fluidGLS sample with:

```
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/fluidsGLS/
$ make
$ ./fluidsGLS
```

DriveWorks Sample Applications

DRIVEWORKS

Sample Applications

DriveWorks is installed on your host PC and DRIVE PX 2 in the `/usr/local/driveworks/` folder.

The HTML Documentation is available in the `/usr/local/driveworks/doc/` folder. Open the `index.html` file in a browser.

Sample binaries are available in the `/usr/local/driveworks/bin` folder. For example, try the DriveNet demo on the DRIVE PX 2:

```
$ cd /usr/local/driveworks/bin  
$ ./sample_drivenet
```

If you have one or multiple cameras plugged in your DRIVE PX 2, you can try the camera replay samples. For example, for one AR0231 RCCB camera:

```
$ ./sample_camera_gmsl --camera-type=ar0231-rccb
```

VisionWorks Sample Applications

VISIONWORKS

Sample Applications

VisionWorks samples can be found in the following directories on the Drive PX 2:

- `/usr/share/visionworks/VisionWorks-*-Samples`
- `/usr/share/visionworks-sfm/VisionWorks-SFM-*-Samples`
- `/usr/share/visionworks-tracking/VisionWorks-Tracking-*-Samples`

You can test them after compiling with `sudo make` in the appropriate folder: the samples will be built in the respective `bin/aarch64/linux` directory.

The HTML Documentation is available in the `/usr/share/visionworks/docs/` folder. Open the `index.html` file in a browser.

Sample sources are available in the respective `sources/` folder.

Inter-Tegra Communication

INTER-TEGRA COMMUNICATION

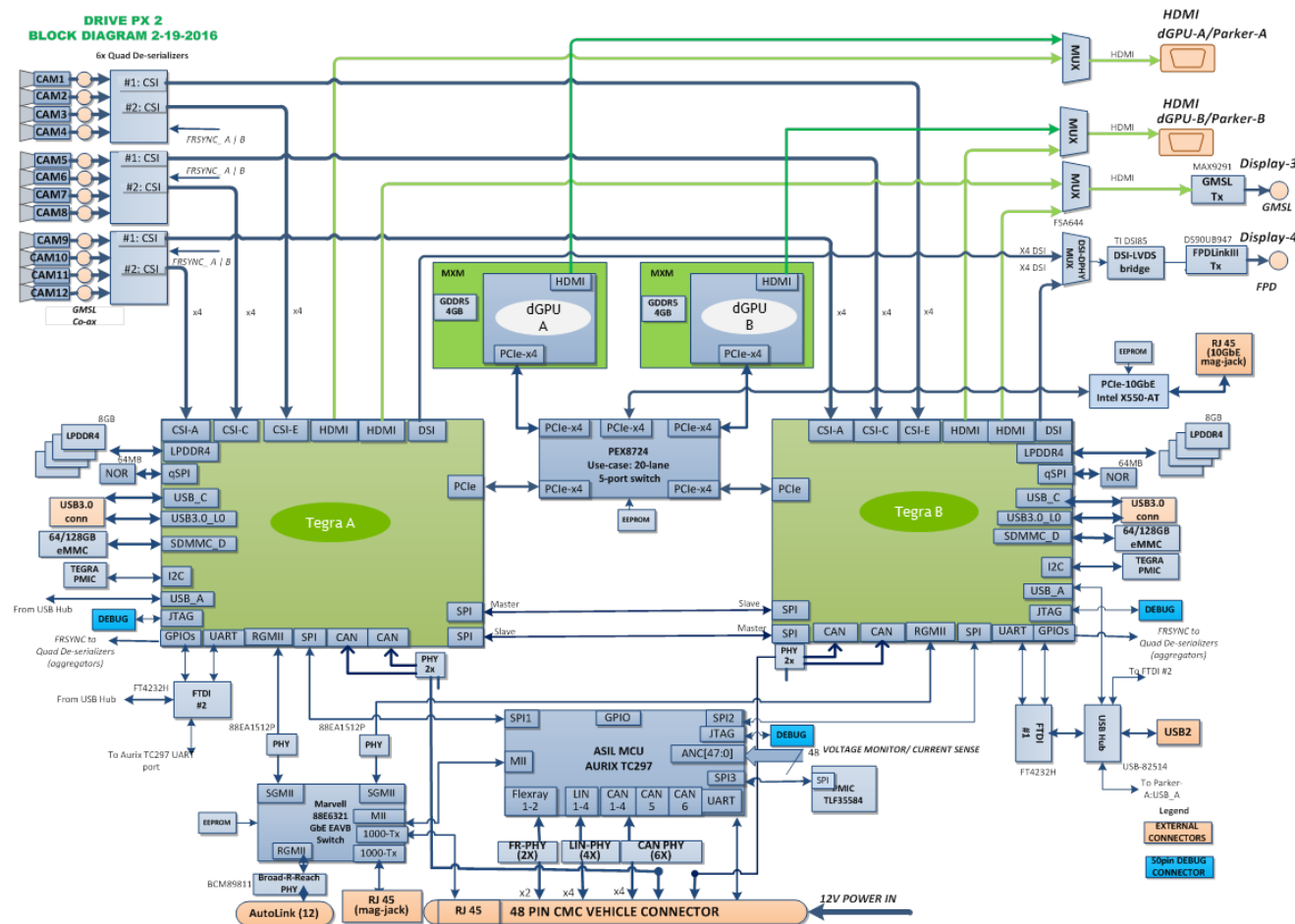
Gb Ethernet Connection

Both Tegra's are inter-connected with Ethernet over a Marvell Gb Switch. They can communicate through this network connection.

If you have a DHCP server in your network, each Tegra is dynamically assigned an IP address at boot. Network configuration is managed by the **systemd-networkd** daemon.

You can then use SCP to copy files between Tegra's, use the socket API to have application communicate between the 2 Tegra's etc.

Note: The Tegra's also have a virtual network interface on eth0:0. It is set to a static address for Tegra-AURIX communications. A conflict may happen if multiple DRIVE PX 2 are located in the same network. See the V4.1L release notes for more information.



DRIVE PX 2 Tips

TIPS

Date & time synchronization

Date & time synchronization on the Tegra's is managed by the the **systemd-timesyncd** daemon. Other alternatives are possible:

- With gPTP: see the DRIVE PX 2 documentation & release notes
- With `ntpdate` for just a 1-time sync:

```
$ sudo apt-get install ntpdate  
  
$ sudo ntpdate 0.ubuntu.pool.ntp.org
```

TIPS

Saving & restoring a list of packages

When migrating to a new Vibrante version, you can save the list of Ubuntu packages installed on your current version and then reinstall them easily on the new one:

- First, get a list of packages installed on the DRIVE PX 2 and save it to a text file:

```
$ dpkg --get-selections | grep -v deinstall > packages_list.txt
```

- Backup the `packages_list.txt` file, install your new Vibrante version on your DRIVE PX 2 and then copy that file back to your new system. You can now restore the complete list of packages with:

```
$ sudo dpkg --set-selections < packages_list.txt
```

```
$ sudo apt-get -u dselect-upgrade
```


QUESTIONS / FEEDBACK

If you have an issue or would like to report feedback, please visit [DevTalk.Nvidia.com](https://devtalk.nvidia.com)'s DRIVE Platforms forum.

You can create new topics, ask questions, and our Solution Architects will respond periodically.

Thank you!

THANK YOU

